Reality v14.0

MultiValue Migration

NECSWS.COM

Orchestrating a brighter world

Document control

Software Version	Document Status	Document Revision	Issue Date	Reason for Change
v14.0	Published	v0.1	March 2008	Final draft

Table of Contents

Section 1: Introduction

Reality was the first commercial release of the MultiValue PICK Database in the early 1970s. Initially, other versions that came onto the market were comparable in features and functions, but over time changes have occurred and a migration process is now required to transfer to Reality from the other versions. This document gives an outline of this process.

A significant area of on-going Reality development concerns improving compatibility with other MultiValue platforms and the migration process. Please contact us before changing your application to work with Reality.

The latest software, patches, migration information and on-line documentation are available at https://reality.necsws.com/ – in the heading click the [Products] tab.

We suggest that you should read these notes through before commencing a migration, as some areas may not apply and usage depends on the way the application has been written and what features have been used. Please note this is a migration process, not a conversion routine.

This document assumes that Reality is being used on Windows. For details of how to install and use Reality on Sun and other UNIX platforms refer to the Installation Guides provided on the Reality Installation CD and in the Reality on-line documentation.

1.1 Main Features

All MultiValue systems share main features that are very similar, but these are sometimes given different names. The following are the names used in Reality:

- Command processor – TCL (Terminal Control Language).
- Query language – English.
- Programming language – DataBasic.
- Batch processing language – Proc

1.2 Reality Documentation

Always refer to the on-line documentation for complete details of features and implementation of Reality Systems.

You will find compatibility information for the current release in the on-line documentation under L<MultiValue compatibility>.

General FAQs can be found on the Reality website – click the [Support] tab and then refer under FAQs. More information about MultiValue compatibility and migration is also available on the web site – click the [Products] tab and refer under Migration.

1.3 Conventions

The following conventions are used in this documentation:

Example	Meaning
TEXT	Bold text in syntax descriptions represents characters typed exactly as shown. For example: **WHO**
{Braces}	Braces enclose options and optional parameters. For example in **BLIST** *file-specifier item-list* {(*options*} • *file-specifier* and *item-list* must be supplied. • one or more single-letter options can be included, as defined for the command; these must be preceded by an open parenthesis, can be given in any order, and are not separated by spaces. Any number of options can be used except where specified in text.
Document Title	Italic text in this font indicates topic titles and other referenced documents. References shown in blue or purple are hyperlinks – if clicked, they will take you to the referenced document or section (purple links are those that you have already visited).
Text	Characters or words in italics indicate parameters which must be supplied by the user. For example in **LIST** *file-specifier* the parameter *file-specifier* is italicised to indicate that you must specify an actual file defined on your system.
[param | param]	Parameters in syntax descriptions shown separated by vertical lines within square brackets indicate that at least one of these parameters must be included. For instance, [**THEN** *statements* | **ELSE** *statements*] indicates that a **THEN** clause or an **ELSE** clause (or both) must be included.
[param || param]	Parameters in syntax descriptions shown separated by pairs of vertical lines within square brackets indicate that one, and only one of these parameters must be included. For instance, [**ON** || **OFF** || **CLOSE**] indicates that just one of the keywords **ON**, **OFF** or **CLOSE** must be included.
…	In syntax descriptions, indicates that the parameters preceding can be repeated as many times as necessary.

Example	Meaning
CTRL+X	Two (or more) key names joined by a plus sign (+) indicate a combination of keys, where the first key(s) must be held down while the second (or last) is pressed. For example, CTRL+X indicates that the CTRL key must be held down while the X key is pressed.
RETURN	Small capitals show key names.
Enter	To enter means to type text then press RETURN. For instance, 'Enter the WHO command' means type **WHO**, then press RETURN. In general, the RETURN key (shown as ENTER or ↵ on some keyboards) must be used to complete all terminal input unless otherwise specified.
Press	Press single key or key combination, but do not press RETURN afterwards.
L<>	This symbol preceding text enclosed in chevrons is used for references to the index in the Reality on-line user documentation (select the Index link in the documentation banner). Enter the text between the chevrons into the field at the top of the Index tool and then double click or press RETURN to display the topic.

Section 2: The Migration Process

In order to migrate to Reality from another MultiValue system you will need the following:

- A Reality V14.0 CD or download with the latest patches. If sufficient for your purposes, you can install a demonstration copy on Windows or UNIX with only three users. You can increase the number of users for a 30-day period by registering with NEC via the NEC web site – https://reality.necsws.com/. If you require more than three users for a longer period, contact NEC.

  The latest software, patches, migration information and on-line documentation are available on the NEC web site's Reality home page – in the heading click the Products tab.

- A host system that meets the requirements given in the Reality Release Information document (available on the on the CD or in the on-line documentation – L<Release information>).

2.1 First Steps

2.1.1 Save the Accounts you want to Migrate

To migrate your data to Reality, you must save it to a medium/format that is compatible with the Reality tape commands. Reality can load data from the following:

- DAT, DLT, 8mm Exabyte and QIC tapes in R83, SMA standard or old Reality format. Your Reality host must have a suitable tape drive.

  Note
  On UNIX, Reality also supports ½ inch tape.

- Tape image files in MultiValue pseudo-floppy format.

  Note
  These must be converted to Reality tape images using the FLOPPYTOTAPE command (see Converting a Pseudo-floppy Save).

Your tape or pseudo-floppy file can contain an account-based save (created with FILE-SAVE for example) or a T-DUMP. The section Load the Data describes how to load these.

2.1.2 Install Reality

To install the demonstration/evaluation version of Reality on Windows, do the following:

1. Log on to Windows as an administrator.
2. Insert the CD-ROM in the drive.
3. When the introductory dialog appears, click the Demo button.
4. If you are installing for the first time, the Reality licence agreement is displayed. You must accept this agreement before you can continue.
5. When prompted, choose the drive on which to install Reality. The software will be installed in the folders RealMan, RealityDemo and RealWebDemo on this drive.

6. When prompted, log off and back on again. Do not remove the CD from the drive.
7. Once you have logged on again, the installation will continue automatically, installing the Reality demonstration database, the RealWeb demonstration and the Reality On-line Documentation.
8. When a message appears to tell you that the installation is complete, click Exit to finish.

For an installation on UNIX or a full installation on Windows, follow the instructions in the appropriate Reality Installation Guide. You will need a set of software keys – contact NEC.

2.1.3 Configure Reality

There are two Reality parameters that you should set before creating your first database.

- Reality uses an enhanced hashing algorithm that is incompatible with that used by some other MultiValue systems. Before creating any databases, you should set the default hash type to the older version.

  Note
  If you use the default Reality hashing algorithm, it will take longer to re-hash items when they are initially restored onto Reality. Also, the item-ids will hash to different groups, so file sizing could be different.

- By default, Reality uses a 1Kbyte frame size. For optimum performance, you should change this to that of your source database.

Both these parameters can be set by editing the file drive:\Realman\14.0\files\config (where drive is the drive where you have installed Reality). Change the value of the HashType parameter to 1 and the FrameSize value to that of your source database (in kilobytes – 1, 2, 4 or 8).

Note
Reality is preconfigured to use DAT or DLT tape. If your data is on 8mm Exabyte or QIC tape, you must configure the appropriate type of tape drive – refer to L<Tape,definitions> for details.

2.1.4 Create a Database

When you install the demonstration version of Reality, a 100Mb database is created. If this is not large enough for your purposes, follow the instructions below.

From a Windows command prompt enter the following commands:

1. If necessary, go to the folder where you want to create your database. Enter:

   cd *path*

   Note
   On Windows Vista, databases that will be available on the network must be created in a folder with unrestricted access, such as a dedicated subfolder within the root folder. A database in a user's home folder cannot be accessed remotely.

2. Use the mkdbase command to create a database:

   mkdbase -N *dbname* **–S***size*

   where

- dbname is the name of the database. A subfolder with this name to hold the database is created in the current folder.

- size is the size of the database in gigabytes (G) or Megabytes (M). For example, using the option -S2G will create a 2Gbyte database consisting of 10 x 200Mb sections (a multi-section database can easily be extended by adding more sections of the same size).

Note

All input is case sensitive – there are lower case n and s options that have different effects.

3. When a database is first created, only the user who created it (the database owner) can log in. To allow other users to log in, unlock the database by entering:

   unlockdbase *dbname*

   The database can also be unlocked from the Reality TCL prompt with the **ENABLE-LOGONS** command.

4. Set the default database. Enter

   netadmin

   and then select option 8. Defaults from the menu, followed by option 1. Default Database. Then select required default database from the list.

   Return to the Windows command prompt by entering q as many times as necessary.

2.2 Logging on to your Database

When you create a new database, a Reality user-id with SYSMAN privileges is created for the database owner (the Windows user who created it). The owner of the database can log on even when the database is locked, without specifying a user-id.

You can log on to the SYSMAN account of the default database as the database owner by entering

reality

at the Windows command prompt. You will not be prompted for a username and password.

Note

- You can also connect to your database via telnet, by entering
  telnet *hostname*
  where *hostname* is the name of the host running Reality (use **localhost** for the local default database). You will be prompted for a username and password.
- To log into other Reality databases, use reality –d dbname from a Windows command prompt, or from the Logon prompt, enter username@dbname.
- By default telnet port 23 is used for incoming connections to Reality. This can be changed using netadmin option, 3. Listening Ports (for details, see L<Listening entries> in the on-line documentation).

2.3 Set up the Database

Start by logging on as the database owner – by default you are logged on to the SYSMAN account

2.3.1 Default User

Reality provides Operating Environments – a way of changing users' settings to simplify migration from a different type of MultiValue system. Several predefined Environments are provided, which you can use either directly or as templates for creating your own Environments.

The environments you create can be applied at TCL when needed with the <SET-ENVIRONMENT command>, or associated with user profiles so that each user is given a suitable profile at log on. You can also set and clear the individual environment options with the L<SET-OPTION> and L<CLEAR-OPTION> commands respectively.

The simplest way to set your users' operating environments is to create a default user that uses an environment that matches that of the MultiValue system from which you are migrating. Three predefined MultiValue environments are currently provided: D3, mvBase and mvEnterprise, or you can create your own environment based on one of these (see L<SSM,environment settings>). The users created when you convert the loaded accounts will be based on the default user.

To set up the default user, log on as the database owner and run the SSM command. Then choose option 2 (Define User Profiles) from the menu and enter D to configure the default user. Enter 23 to set the environment, followed by the name of the environment you require; you can list the available environments by entering ?.

For more details, refer to L<User profile> and L<Operating Environment>

2.4 Load the Data

Once you have created a correctly configured database, you can load the data from your source system.

2.4.1 Converting a Pseudo-floppy Save

The pseudo-floppy (.vtf) format used by many MultiValue systems is not compatible with Reality. If your data is in a pseudo-floppy file, before it can be loaded into your database, it must be converted into a format that can be read by the Reality tape commands.

If you have not already done so, log on as the database owner. Then, to convert your pseudo-floppy save, enter:

FLOPPYTOTAPE *path filename* (*options*

where path and filename specify the location and name of the file containing the pseudo-floppy image. The converted file is saved in the same host folder, and is given the same name as the pseudo-floppy image, but with the file extension '.rti' (uncompressed) or '.rci' (compressed).

Note

- If you prefer, you can save the converted image in a different folder by specifying the (A option – you will then be prompted for the path of the required save folder.
- By default, the converted image will be uncompressed. If you prefer to use a compressed mage, specify the (C option.
- FLOPPYTOTAPE can currently convert D3, mvEnterprise and jBASE pseudo-floppy files. Options are available to specify the particular file type.
- See L<FLOPPYTOTAPE command> in the on-line documentation for details of these and other options.

2.4.2 Loading MultiValue Accounts

To load the accounts from your tape or converted tape image, ensure that you are in the SYSMAN account and do the following:

1. If you are loading from a pseudo-floppy save, set a pointer to the converted Reality Tape Image file:

   T-DEVICE 1 *path*

   where path is the location and file name of the converted tape image. This command temporarily configures tape device 1 to use the specified tape image file.

2. Attach the device:

   T-ATT 1 SIZE=*recordSize*

   where recordSize is the record (block) size used when the data was saved. Refer to <T-ATT command> for details.

   Note

   The number following the command name (1 in this case) identifies the tape device. Reality is supplied with device 1 preconfigured for a DAT or DLT tape. If you are using 8mm Exabyte or QIC tape, substitute the number of the device you have configured – see Configure Reality.

3. Rewind the tape file:

   T-REW 1

4. Position the tape to the start of the data. You can find out whether the tape is correctly positioned with the L<T-READ> TCL command. The first data section block should be similar to one of the following:

   ```
   Record =1 ; 16000 bytes
         0 _F^0^0001^..9=...?SYSTEM^DL^20480^23]B0 23,1^1^^SY
   Record =1 ; 8192 bytes
         0 _D^1^1^0036SYSMAN^D^20480^29^1^^SYSUPDATE^^SYS2(25
   ```

   If necessary, repeat the T-READ. Once you have identified the correct position, you can use the L<T-BCK command> to return the tape to the correct position.

   Note

   The T-BCK command cannot be used on quarter-inch cartridge ((QIC) drives.

5. Restore the accounts

   ACCOUNT-RESTORE *

   If the data cannot be loaded, contact NEC for assistance.

If you have any problems loading your data, please contact NEC for assistance.

Note

- In general, ACCOUNT-RESTORE will exclude data that is in an unrecognised format. Since each MultiValue system uses its own format for indexes, binary items and items larger than 32Kb, these will usually be excluded from the loaded data. You can recreate any indexes once your accounts have been converted (see Indexes for details) and binary items generated by the compiler can be recreated by recompiling (see Recompiling MultiValue Programs). For assistance with loading other binary items and items larger than 32Kb, contact NEC.
- If you are unable to load your data, you may be able to exclude indexes, binary items and items larger than 32Kb from the save – refer to the documentation for your MultiValue system. NEC are happy to assist and would appreciate any feedback about such problems.

2.4.3 Convert the Loaded Accounts

Caution

This must be done before attempting to log on to any of the loaded accounts.

To convert the loaded accounts, use the MIGRATE.ACCOUNT utility. Ensure that you are in the SYSMAN account and enter:

MIGRATE.ACCOUNT *accountList*

where accountList is a space-separated list of account names. You are prompted for the type of MultiValue system from which the accounts were saved. If the system from which you are migrating is not listed, choose the MULTIVALUE option. Refer to <MultiValue compatibility,MIGRATE.ACCOUNT> for full details.

Each account is processed as follows:

- The account password is removed.
- The account privilege level is converted to upper case. SYS3 account privilege is converted to SYS2
- Attribute 1 of the D-pointer in the SYSTEM file is processed to remove invalid options or convert them to the appropriate Reality options. See <Account,definition item> for details.
- Provided one does not already exist, a user-id is created with the same name as the migrated account. If this is a local or remote account, the new user's default account is set to the migrated account; if it is a synonym (Q-pointered) account, the default account is set to the referenced local or remote account.

The following changes are made only to local accounts:

Files with names that are the same as items in the appropriate data sections of the

NEWAC file are renamed with the prefix NON-NEC-.

- Items with the same names as items in the NEWAC file are moved to the default data section of the file NON-NEC-MD-ARCHIVE.
- The contents of the appropriate data sections of the NEWAC file are copied into the MD. Items in the USER-DELETE and USER-ADD lists are processed appropriately.
- Attribute 1 of each D-pointer (whether in the MD of the account or in a file dictionary) is processed to remove invalid options or convert them to the appropriate Reality options. For details of valid options see <File,definition item> and <Data level descriptor>.
- If converting from D3, index references in attribute 8 of each D-pointer are removed (you will need to recreate your indexes once conversion is complete – see Indexesfor more information).
- If converting from jBASE, carriage return/line feed and line feed characters in non-binary items are replaced with attribute marks. Leading spaces are removed from attributes in PQ and PQN items.
- If converting from mvEnterprise, directory pointers (dictionary items with attribute 1 = "QR") are replaced by Reality directory views (see L<Directory view>).

Note

If the directory specified does not exist, an error message will be displayed when you attempt to access the associated Reality file.

- The original versions of MD and dictionary items, other than D-pointers, that have been modified by the conversion process are saved in the PROCESSED data section of the file NON-NEC-MD-ARCHIVE. Items that could not be processed remain in the default data section of this file.

A processing log is saved in the file MIGRATION-LOG in the SYSMAN account.

2.4.4 Loading a T-DUMP

T-DUMP is used to save selected items from a file. When loading the data onto a Reality database, you must first create a file to hold the data:

1. If you have not already done so, log on to the database as the database owner.
2. Create an account to hold your file by entering

   CREATE-ACCOUNT

   You will be prompted for the following:

   - Whether the account should be Local, Remote or Q-pointered. Enter L.
   - The account name.

   For the remaining prompts, you can press RETURN to accept the default. Most of them can be changed later if required – see <Local account creation> and <Account,definition item> for details.
3. Log to the new account by entering.

   LOGTO *accountName*
4. Log to the new account by entering.

   CREATE-FILE *filename*
5. If you are loading from a pseudo-floppy save, set a pointer to the converted Reality

   Tape Image file:

   T-DEVICE 1 *path*

   where path is the location and file name of the converted tape image. This command temporarily configures tape device 1 to use the specified tape image file.
6. Attach the tape device:

   T**-ATT 1 SIZE**=*recordSize*

   where recordSize is the record (block) size used when the data was saved.
7. Rewind the tape file:

   T-REW 1
8. Use the <T-FWD command> as many times as necessary to position the tape at the start of the data you wish to load.
9. Load the data by entering

   T-LOAD *filename*

> ***Note***
> any dictionary items must be loaded separately with
> **T-LOAD DICT** *fileName*

2.5 Using Reality

2.5.1 Logging on to Loaded Accounts

Caution

Do not log on to a loaded account until it has been converted using MIGRATE.ACCOUNT.

Until such time as you have recompiled your applications, when logging to your loaded accounts you should use the **C** option:

LOGTO *accountName* (**C**

This prevents execution of any Logon Proc that might attempt to run a cataloged program.

2.5.2 Terminal Emulation

Different terminals have their own characteristics. In Reality, a number of these are predefined, including NEC's own PRISM emulation.

The TDM (<Terminal Definition Maintenance>) command allows you to define additional terminal types and to check to see which are already configured. Check the terminal type being used by entering the <TERM command> at TCL. You can change to a different terminal type by entering:

TERM ,,,,,,,*typeNumber*

that is, enter 8 commas (for the 8th parameter) followed by the terminal type number.

You can set the default terminal type with the SSM utility – select option 1 (Network File Maintenance) and enter D at the PLID prompt. Then set option 8 (terminal type) to the required type number. See <SSM,NETWORK file maintenance> for details.

2.5.3 Reality Editors

Reality includes several editors which you can use to create and modify file and dictionary items. The most commonly used are:

- The terminal line editor – L<EDIT command>.
- The terminal screen editor – L<SEDIT>.
- The Windows editor, L<RealEdit>. This must be separately installed on a PC.

2.6 Indexes

Since each MultiValue system uses its own format for indexes, these will usually be excluded when you load your data. To recreate your indexes, use the <CREATEINDEX> TCL command, specifying the file (data section) name and the required dictionary definition. You can also create an index on the item-ids in a file.

2.7 Recompiling MultiValue Programs

The differences between Reality and MultiValue systems mean that you will not be able to simply run your compiled applications, and you may need to change your code before you can successfully recompile (using the L<BASIC command>). After recompilation, all applications must be recataloged with the L<CATALOG command>, using the O option to ensure that any existing command definition items are overwritten.

Reality includes a number of features that simplify this process (see below).

When you have recompiled and recataloged all your applications, log on to the SYSMAN account and run L<UPDATE-ACCOUNT> or L<UPDATE-ALL-ACCOUNTS> to restore any Reality system commands that might have been overwritten (you may need to change the names of your program modules to ensure that they do not overwrite Reality system commands).

2.7.1 Compiler Version

The evaluation version of Reality is configured to use the latest DataBasic compiler, but for backwards compatibility, a commercial installation will use the version supplied with Reality V11.0. If you have upgraded a previous version or installed Reality with a set of software keys and want to take advantage of the latest DataBasic features, edit the file item /SYSFILES/BASIC-COMPILERS BASIC*DEFAULT and change attribute 2 from `BASIC*11` to `BASIC*14`.

You can also use the BASIC command with the C option to specify a compiler other than the default.

2.7.2 $OPTIONS Statement

This allows the Reality DataBasic compiler to emulate that on another MultiValue system (D3, mvBase and mvEnterprise settings are currently available). This can reduce or eliminate the changes needed to your source code. See L<MultiValue compatibility, $OPTIONS statement>

The $OPTIONS statement can also be used to change the types, names and locations of the items generated by the DataBasic compiler, to match other MultiValue systems. If the statement

```
$OPTIONS EXEC.OBJ
```

is added at the beginning of a code module, the compiler will generate an executable (platform-specific) item in the dictionary of the file. This dictionary item will have the same name as the source item. See L<$OPTIONS statement> for more details.

Note

this option is set by selecting a MultiValue Operating Environment – see Default User.

2.7.3 AutoInclude

If you have a large number of code modules, adding a $OPTIONS statement to the beginning of each one can be a major task. To simplify migration, you can place this in an auto-include item. When your programs are compiled, the contents of this item will be automatically included at the beginning of your code.

- If you create an item called #AUTOINCLUDE in your source file, the contents of this item will be automatically included at the beginning of every code module in the same data section.
- If you create an item called #AUTOINCLUDE in the default data section of the file BASIC-COMPILERS in the SYSFILES account (edit the item /SYSFILES/BASIC-COMPILERS #AUTOINCLUDE – this will create a new item if it does not already exist), its contents will be included in every DataBasic program in your database after any local automatically included code.

See <Automatically including common code>.

2.7.4 @ Function

The meanings of the terminal control codes that you pass to the <@ function> vary between MultiValue systems. If you set a compatibility mode with the $OPTIONS statement, many of these codes will automatically be mapped to their Reality equivalents. However, if you are migrating from a MultiValue product that is not currently supported, you may find that the codes you are using do not have the expected effect.

Although it is possible to change your application to use the Reality codes, a simpler solution can be to use TDM to create a new custom terminal type (see above). For example, if you are using VT220 compatible terminals, create a new terminal type based on VT220. Then, for each @ code used in your application, change the output sequence to that which produces the required effect.

Refer to <Terminal Definition Maintenance> for details.

2.7.5 Line Numbering

When a runtime error occurs in a DataBasic program, the line number given in the error message will take into account items that have been included (using the <INCLUDE statement> and autoinclude) and chained (using the L<$CHAIN statement>).

If you list the program using the <BLIST command> with the **M** option the first column contains the source line number, while the second column contains the expanded line number, taking into account included and chained items (that is, the one given in the error message). Look up the line number from the error message in the second column – the first column will give you the corresponding line in the source item.

If you cannot find the line number you require in the second column, the error will have occurred in an included or chained item. Relist using BLIST with the **ME** options to expand included and chained items.